

EME*: extending EME to handle arbitrary-length messages with associated data

(Preliminary Draft)

Shai Halevi*

May 18, 2004

Abstract

We describe a mode of operation EME* that turns a regular block cipher into a length-preserving enciphering scheme for messages of (almost) arbitrary length. Specifically, the resulting scheme can handle any bit-length, not shorter than the block size of the underlying cipher. In addition, it handles associated data of arbitrary bit-length. EME* is a refinement of the EME mode of Halevi and Rogaway, and it inherits the efficiency and parallelism from the original mode.

1 Specification of EME* Mode

We construct from block cipher $E: \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ an enciphering scheme with associated data that we denote by $\text{EME}^*[E]: (\mathcal{K} \times \{0, 1\}^{2n}) \times \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$, where \mathcal{K} is the same as the underlying cipher, $\mathcal{T} = \{0, 1\}^*$, and $\mathcal{M} = \{0, 1\}^{n+}$. In words, the key for enciphering scheme $\text{EME}^*[E]$ consists of one key K of the underlying block cipher and two n -bit blocks, L and R . $\text{EME}^*[E]$ accepts messages of any bit length greater than or equal to n , and associated data of arbitrary bit-length.

The scheme $\text{EME}^*[E]$ follows the same general principles of the tweakable scheme EME from [3]. Roughly, it consists of two layers of masked ECB encryption, with a layer of “lightweight mixing” in between. A complete specification of the enciphering scheme $\text{EME}^*[E]$ is given in Figure 1, and an illustration (for a message of less than n^2 bits) is provided in Figure 2. For those familiar with EME, the differences between EME and EME^* are as follows:

- *More than one mask.* The EME scheme uses (multiples of) a single mask value M in the “lightweight masking” layer. It was shown in [3], however, that this masking technique with just one mask cannot be used for messages longer than n^2 bits.

To handle longer messages we adopt the approach that was proposed in the appendix of [3], breaking the message to chunks of at most n^2 bits each, and computing a different mask value for every chunk. As we explain below, we use yet another mask to handle the last partial block (if needed).

*IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA, shaih@watson.ibm.com
<http://www.research.ibm.com/people/s/shaih/>

<pre> function $F_{K,R}(T_1 \dots T_{\ell-1}, T_\ell)$ $T_1 = \dots = T_{\ell-1} = n, 0 < T_\ell \leq n$ 00 if T is empty return $E_K(R) \oplus R$ 10 for $i \in [1.. \ell - 1]$ do 11 $TT_i \leftarrow 2^i R \oplus T_i$ 12 $TTT_i \leftarrow E_K(TT_i) \oplus 2^i R$ 20 if $T_\ell = n$ then 21 $TT_\ell \leftarrow 2^\ell R \oplus T_\ell$ 22 $TTT_\ell \leftarrow E_K(TT_\ell) \oplus 2^\ell R$ 23 else $TT_\ell \leftarrow (2^\ell + 1)R \oplus (T_\ell 10..0)$ 24 $TTT_\ell \leftarrow E_K(TT_\ell) \oplus (2^\ell + 1)R$ 30 return $TTT_1 \oplus \dots \oplus TTT_\ell$ </pre>	<pre> Algorithm $\mathbf{E}_{K,L,R}(T; P_1 \dots P_m)$ $P_1 = \dots = P_{m-1} = n, 0 < P_m \leq n$ 101 $H \leftarrow F_{K,R}(T)$ 102 if $P_m = n$ then $lastFull \leftarrow m$ 103 else $lastFull \leftarrow m - 1$ 104 $PPP_m \leftarrow P_m$ padded with 10..0 110 for $i \leftarrow 1$ to $lastFull$ do 111 $PP_i \leftarrow 2^{i-1}L \oplus P_i; PPP_i \leftarrow E_K(PP_i)$ 120 $SP \leftarrow PPP_2 \oplus \dots \oplus PPP_m$ 121 $MP_1 \leftarrow PPP_1 \oplus SP \oplus H$ 122 if $P_m = n$ then $MC_1 \leftarrow E_K(MP_1)$ 123 else $MM \leftarrow E_K(MP_1); MC_1 \leftarrow E_K(MM)$ 124 $C_m \leftarrow P_m \oplus (MM \text{ truncated})$ 125 $CCC_m \leftarrow C_m$ padded with 10..0 126 $M_1 \leftarrow MP_1 \oplus MC_1$ 130 for $i = 2$ to $lastFull$ do 131 $j = \lceil i/n \rceil, k = (i - 1) \bmod n$ 132 if $k = 0$ then 133 $MP_j \leftarrow PPP_i \oplus M_1; MC_j \leftarrow E_K(MP_j)$ 134 $M_j \leftarrow MP_j \oplus MC_j$ 135 $CCC_i \leftarrow MC_j \oplus M_1$ 136 else $CCC_i \leftarrow PPP_i \oplus 2^k M_j$ 140 $SC \leftarrow CCC_2 \oplus \dots \oplus CCC_m$ 141 $CCC_1 \leftarrow MC_1 \oplus SC \oplus H$ 142 for $i \leftarrow 1$ to $lastFull$ do 143 $CC_i \leftarrow E_K(CCC_i); C_i \leftarrow CC_i \oplus 2^{i-1}L$ 150 return $C_1 \dots C_m$ </pre>
--	---

Figure 1: Enciphering under $\mathbf{E} = \text{EME}^*[E]$, where $E: \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a block cipher. The associated data is $T \in \{0, 1\}^*$, the plaintext is $P = P_1 \dots P_m$ and the ciphertext is $C = C_1 \dots C_m$. The deciphering procedure \mathbf{D}_K^T can be obtained from \mathbf{E}_K^T by replacing $E_K(\cdot)$ by $E_K^{-1}(\cdot)$ everywhere (but without modifying the function $F_{K,R}$).

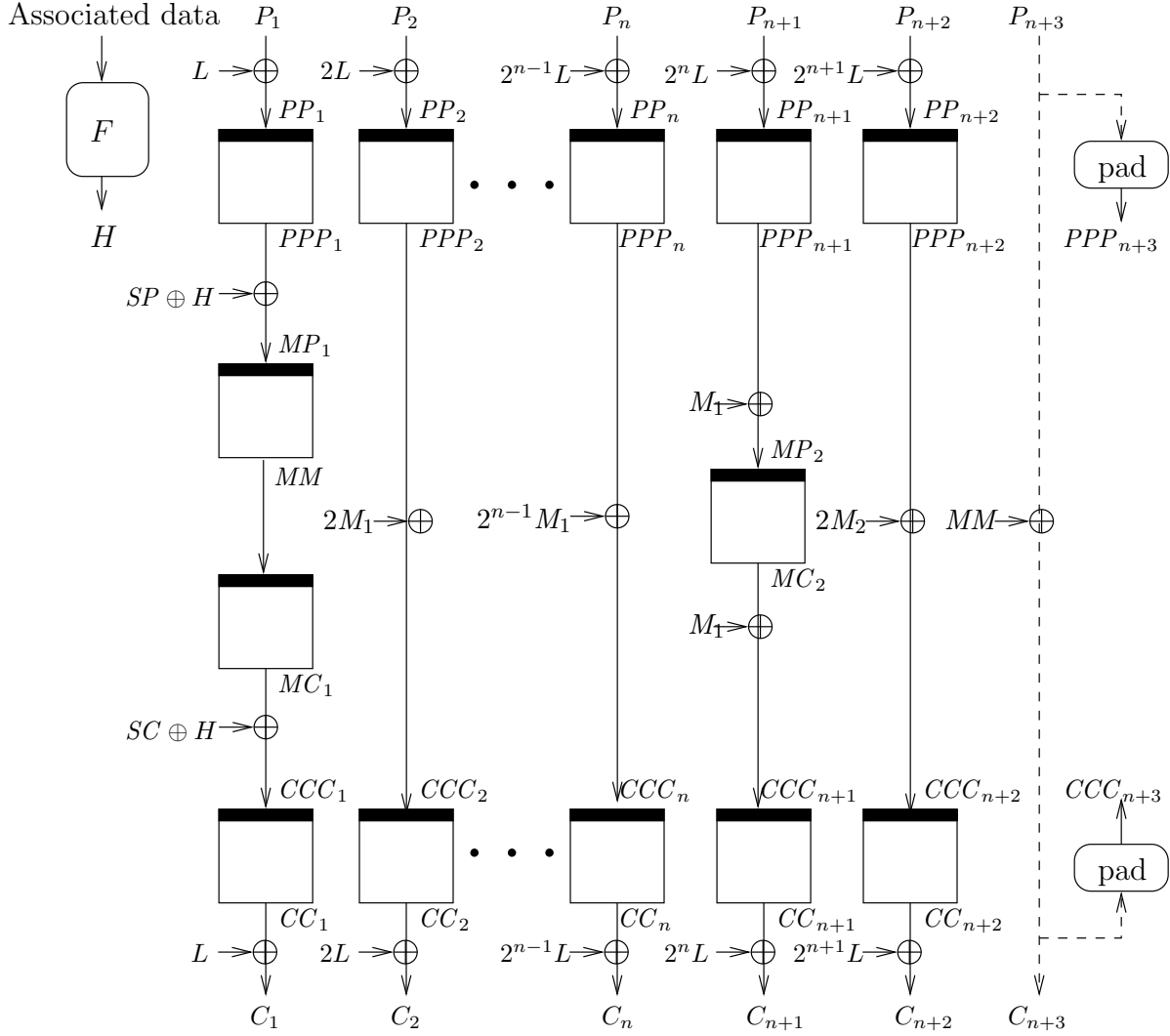


Figure 2: Enciphering under EME* a buffer with $n + 2$ full blocks and one partial block. The boxes represent E_K . We set the masks as $SP = PPP_2 \oplus \dots \oplus PPP_{n+3}$, $M_i = MP_i \oplus MC_i$, and $SC = CCC_2 \oplus \dots \oplus CCC_{n+3}$.

- *Hashing the “tweak”*. The original EME scheme requires that the “tweak value” be an n -bit string, whereas here we allow associated data of any length. For this purpose, we hash the associated data to an n -bit string. The hash function that we use should only be xor-universal, yet we chose to implement it using the underlying block cipher in a PMAC-like mode [1].
- *The partial block*. To handle the last partial block (if any), we compute yet another mask and add it into the last partial plaintext block, thus getting the last partial ciphertext block. Also, we add the partial plaintext block and the “hashed tweak” after the first masked-ECB layer and before computing the first mask, and then we add the partial ciphertext block and the “hashed tweak” again before the second masked-ECB layer.

If needed, one could derive the blocks L, R from the key K , say by setting $L = 2E_K(0)$ and $R = 3E_K(0)$. The only drawback is that this will add a few more pages to the proof of security.

2 Security of EME*

The following theorem relates the advantage an adversary can get in attacking $\text{EME}^*[E]$ to the advantage an adversary can get in attacking the block cipher E .

Theorem 1 [EME* security] Any adversary that tries to distinguish $\text{EME}^*[\text{Perm}(n)]$ from a truly random tweakable length-preserving permutation, using at most q queries totalling at most σ_n blocks (some of which may be partial), has advantage at most $(2q + (2 + \frac{1}{n})\sigma_n)^2 / 2^{n+1}$. Namely,

$$\text{Adv}_{\text{EME}^*[\text{Perm}(n)]}^{\pm\text{prp}}(q, \sigma_n) \leq \frac{(2q + (2 + \frac{1}{n})\sigma_n)^2}{2^{n+1}} \quad (1)$$

Corollary 1 Fix $n, t, q, \sigma_n \in \mathbb{N}$ and a block cipher $E: \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$. Then

$$\text{Adv}_{\text{EME}^*[E]}^{\pm\text{prp}}(t, q, \sigma_n) \leq \frac{(2q + (2 + \frac{1}{n})\sigma_n)^2}{2^{n+1}} + 2 \text{Adv}_E^{\pm\text{prp}}\left(t', 2q + (2 + \frac{1}{n})\sigma_n\right)$$

where $t' = t + O(n\sigma_n)$. □

We note that the theorem and corollary do not restrict messages to one particular length: proven security is for a variable-input-length (VIL) cipher, not just fixed-input-length (FIL) one. The proof of Theorem 1 is given in Appendix A. Corollary 1 embodies the standard way to pass from the information-theoretic setting to the complexity-theoretic one.

References

- [1] J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. In L. Knudsen, editor, *Advances in Cryptology – EUROCRYPT ’02*, volume 2332 of *Lecture Notes in Computer Science*, pages 384–397. Springer-Verlag, 2002.
- [2] S. Halevi and P. Rogaway. A tweakable enciphering mode. In D. Boneh, editor, *Advances in Cryptology – CRYPTO ’03*, volume 2729 of *Lecture Notes in Computer Science*, pages 482–499. Springer-Verlag, 2003. Full version available on the ePrint archive, <http://eprint.iacr.org/2003/148/>.

- [3] S. Halevi and P. Rogaway. A parallelizable enciphering mode. In *The RSA conference – Cryptographer’s track, RSA-CT’04*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304. Springer-Verlag, 2004. Full version available on the ePrint archive, <http://eprint.iacr.org/2003/147/>.
- [4] J. Kilian and P. Rogaway. How to protect DES against exhaustive key search. *Journal of Cryptology*, 14(1):17–35, 2001. Earlier version in CRYPTO ’96. www.cs.ucdavis.edu/~rogaway.

A Proof of Theorem 1 — Security of EME*